**In the Specification:**

Please amend the Abstract of the Disclosure as indicated below.

~~Apparatus and methods of synchronizing resource access for a multi-threaded computing environment are disclosed. The mechanism is operable to manage a sequence of one or more mutexes, wherein the sequence of mutexes is associated with a resource and each mutex may be allocated to one thread. The mechanism is operable, when a requesting thread attempts an access to the resource, to lock a mutex, wherein the locked mutex is allocated to the requesting thread and to attempt to lock a previous mutex in the sequence if present. The requesting thread is suspended if the previous mutex is already locked until the previous mutex is unlocked in response to a previous thread finishing access to the resource. When attempting to lock a previous mutex in the sequence where the previous mutex is unlocked, the mechanism is operable to lock the previous mutex on behalf of the requesting thread and then to unlock the previous mutex on behalf of the requesting thread. The resource access control mechanism unlocks the mutex allocated to the requesting thread in response to the requesting thread completing access to the resource. By ensuring that only one thread is waiting on the release of any one mutex, problems with ordering of the resource access is avoided.~~ A resource access control mechanism for a multi-threaded computing environment associates a sequence of one or more mutexes with a resource. When a requesting thread attempts to access the resource, a mutex is locked and allocated to the requesting thread, and if a previous mutex in the sequence is present, an attempt to lock the previous mutex is made. If the previous mutex is already locked, the requesting thread is suspended until the previous mutex is unlocked.